Fundamentals of R

1. Installation of R & RStudio

What is R?

- R is an open-source programming language used for statistical computing and data analysis.
- It is widely used in data science, machine learning, and statistical modeling.

Installation Steps:

1. **Download R**: Go to the official R website CRAN and download the latest version of R.
2. **Install RStudio**: Download **RStudio**, an IDE for R, from Posit.
3. **Run and Verify**: Open RStudio and type print("Hello, R!") to verify the installation.

---

2. Features of R

- **Open-source**: Free to use and modify.
- **Statistical Computing**: Supports various statistical operations.
- **Graphical Capabilities**: Can create high-quality visualizations.
- **Extensive Libraries**: Thousands of pre-built packages available via **CRAN**.
- **Supports Multiple Data Formats**: Works with CSV, Excel, JSON, databases, etc.
- **Highly Extensible**: Users can develop custom packages.

---

3. Variables in R

- Variables are used to store values in R.
- You can assign values using <- or =.

Example:

r

CopyEdit

```
x <- 10     # Assigning value 10 to x
y = 20      # Assigning value 20 to y
name <- "R Programming"
```

Variable Naming Rules:

✅ Can contain letters, numbers, and underscores.
❌ Cannot start with a number.
❌ Cannot use reserved keywords like if, else, function, etc.

---

4. Constants in R

Constants are fixed values that do not change.

Common Constants in R:

- pi → 3.141593
- TRUE / FALSE → Logical constants
- Inf → Infinity
- NaN → Not a Number
- NULL → Represents an empty or undefined value

Example:

r

CopyEdit

```
print(pi)    # Output: 3.141593
print(TRUE)  # Output: TRUE
print(Inf)   # Output: Inf
```

---

5. Operators in R

Operators perform calculations or comparisons.

Types of Operators in R:

1. **Arithmetic Operators:**
    - + (Addition), - (Subtraction), * (Multiplication), / (Division), ^ (Exponent), %% (Modulus), %/% (Integer division)
    - Example:

r

CopyEdit

```
a <- 10
b <- 3
sum <- a + b   # Output: 13
```

2. **Relational (Comparison) Operators:**
    - > (greater than), < (less than), == (equal to), != (not equal to)
    - Example:

r

CopyEdit

x <- 5

y <- 10

print(x > y) # Output: FALSE

3. **Logical Operators:**
    - & (AND), | (OR), ! (NOT)
    - Example:

r

CopyEdit

a <- TRUE

b <- FALSE

print(a & b) # Output: FALSE

4. **Assignment Operators:**
    - <-, ->, =
    - Example:

r

CopyEdit

x <- 100 # Assigns 100 to x

5. **Miscellaneous Operators:**
    - : (Sequence), $ (Access list elements), %in% (Check element presence)
    - Example:

r

CopyEdit

v <- 1:5   # Creates a sequence from 1 to 5

print(v)

---

6. Data Types and R Objects

Basic Data Types in R:

1. **Numeric**: Decimal numbers (10.5, 3.14)
2. **Integer**: Whole numbers (5L, 100L)

3. **Character**: Text ("Hello", "R Programming")
4. **Logical**: Boolean (TRUE, FALSE)
5. **Complex**: Complex numbers (4+2i)

Data Structures (Objects) in R:

1. **Vector**: One-dimensional data

r

CopyEdit

vec <- c(1, 2, 3, 4)

print(vec)

2. **Matrix**: 2D table with rows & columns

r

CopyEdit

mat <- matrix(1:9, nrow=3, ncol=3)

print(mat)

3. **List**: Stores different data types

r

CopyEdit

lst <- list(name="Alice", age=25, scores=c(90,85,88))

print(lst)

4. **Data Frame**: Table with rows & columns (used for datasets)

r

CopyEdit

df <- data.frame(Name=c("A", "B"), Age=c(21, 25))

print(df)

5. **Factor**: Categorical data

r

CopyEdit

gender <- factor(c("Male", "Female", "Male"))

print(gender)

---

7. Accepting Input from Keyboard

R allows user input using readline() or scan().

Example:

r

CopyEdit

```
name <- readline("Enter your name: ")
print(paste("Hello,", name))
```

- readline() is used for text input.
- scan() is used for numeric input.

Example for Numeric Input:

r

CopyEdit

```
num <- scan()
print(num)
```

---

8. Important Built-in Functions in R

R has many built-in functions for mathematical and statistical calculations.

Mathematical Functions:

- sum(), mean(), median(), sd(), var(), sqrt(), abs(), round(), log(), exp().
- Example:

r

CopyEdit

```
numbers <- c(10, 20, 30, 40, 50)
print(sum(numbers))  # Output: 150
print(mean(numbers)) # Output: 30
```

String Functions:

- paste(), toupper(), tolower(), substr(), nchar().
- Example:

r

CopyEdit

```
text <- "Hello R"
print(toupper(text))  # Output: HELLO R
```

Statistical Functions:

- min(), max(), range(), quantile().
- Example:

r

CopyEdit

```
data <- c(5, 10, 15, 20)

print(max(data))  # Output: 20
```

Conclusion

- **R is a powerful language for data analysis, statistics, and visualization.**
- **Understanding variables, operators, data types, and built-in functions is essential for programming in R.**